



Predição de Séries Temporais Baseada em Redes Neurais Artificiais

Aluno: João Pedro de Oliveira Pagnan [FEEC/UNICAMP]

Orientador: Prof. Levy Boccato [FEEC/UNICAMP]

Coorientador: Prof. Romis Ribeiro de Faissol Attux [FEEC/UNICAMP]

Departamento de Engenharia de Computação e Automação Industrial (DCA)

Faculdade de Engenharia Elétrica e de Computação (FEEC)

Universidade Estadual de Campinas (UNICAMP)

Palavras-chave – Redes Neurais Artificiais, Sistemas Caóticos, Séries Temporais

1. Introdução

A predição de séries temporais é uma das aplicações mais interessantes do tratamento de informação. O desafio de antecipar padrões de comportamento e construir modelos que sejam apropriados para explicar determinados fenômenos da natureza tem importância para a biologia, economia, automação industrial, meteorologia e diversas outras áreas da ciência [1].

Na literatura, encontramos diversos tipos de modelos para a predição de séries temporais, desde métodos clássicos lineares, como o modelo autorregressivo (AR) [1] até métodos não-lineares utilizando, por exemplo, redes neurais artificiais, sendo que dessas se destacam as redes do tipo *Multilayer Perceptron* (MLP) [2] e as redes recorrentes, especialmente a *Long Short-Term Memory* (LSTM) [3] e a *Echo State Network* (ESN) [4].

Uma classe de sistemas dinâmicos particularmente relevante dentro do contexto de modelagem e predição de séries temporais está ligada à ideia de dinâmica caótica. Diversos fenômenos naturais, como a dinâmica populacional de uma espécie, a dinâmica atmosférica de uma região, ou até mesmo as órbitas de um sistema com três ou mais corpos celestes podem exibir comportamento caótico. Apesar de serem determinísticos (e, portanto, previsíveis), esses sistemas são extremamente sensíveis às condições iniciais [5]. Isso causa um problema para a predição das séries temporais originadas por eles, pois uma pequena incerteza na medida afetará toda a previsão.

Tendo em vista o desempenho de modelos não-lineares para previsão de diversas séries temporais [3], optamos por estudar a aplicabilidade de redes neurais artificiais à previsão de séries relacionadas a sistemas com dinâmica caótica.

Esta pesquisa comparou o desempenho de quatro arquiteturas distintas de redes neurais artificiais: a rede *Multilayer Perceptron* [2], a rede *Long Short-Term Memory* [3], a rede *Gated Recurrent Unit* (GRU) [6] e, por fim, a rede *Echo State Network* (ESN) [4].

A comparação foi realizada em quatro cenários de sistemas caóticos, sendo dois destes a tempo discreto e dois a tempo contínuo. No caso, os sistemas a tempo discreto foram as séries temporais do mapa de Hénon [7], e a série temporal relacionada ao mapa logístico [8]. Já os sistemas a tempo contínuo foram o sistema de Lorenz [9] e as equações de Mackey-Glass [10]. Nos sistemas multidimensionais, como o mapa de Hénon e o sistema de Lorenz, foi utilizada a série temporal em \hat{x} .

Iniciamos a análise através de um processo de *grid-search* para determinarmos os parâmetros ótimos para as redes neurais em cada cenário. Em seguida, utilizando os melhores parâmetros, realizou-se um estudo da progressão erro quadrático médio (EQM) com o número de amostras de entrada do modelo preditor (nesse caso, chamado de K). Por fim, comparamos qual foi a média e o desvio padrão do EQM com o melhor valor de K de cada modelo nos quatro cenários.

Através desta pesquisa, percebe-se que a ESN possui o melhor desempenho dentre todos os modelos, em todos os cenários. As redes *Echo State Network* têm a vantagem de necessitarem de bem menos recursos computacionais do que os outros modelos avaliados, obtendo um desempenho consideravelmente superior mesmo com um tempo de treinamento ínfimo se comparado às redes neurais artificiais tradicionais.

2. Metodologia

2.1. Geração de dados

Como o intuito do projeto é analisar o desempenho dos modelos preditores citados em sistemas caóticos, os dados necessários para o treinamento e para a análise da *performance* podem ser obtidos através da solução numérica dos sistemas caóticos. As próximas seções evidenciam o processo utilizado, junto com os parâmetros escolhidos para os sistemas caóticos.

2.1.1. Mapa de Hénon

O mapa de Hénon foi um dos sistemas a tempo discreto escolhidos para esta pesquisa. Esse sistema foi proposto por

Michel Hénon em 1976 como um modelo simplificado de uma seção de Poincaré do atrator de Lorenz, sendo descrito pelas equações abaixo [7]:

$$x[n+1] = y[n] + 1 - a \cdot (x[n])^2 \quad (1a)$$

$$y[n+1] = b \cdot x[n] \quad (1b)$$

Para esta pesquisa, foram utilizados os valores usuais para os parâmetros a e b . Logo, têm-se que $a = 1.4$ e $b = 0.3$. Além disso, neste e nos outros sistemas, foi gerado um conjunto de 5000 dados. As figuras a seguir mostram a série temporal em \hat{x} e o atrator obtido com a simulação:

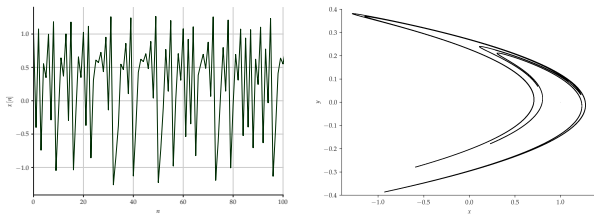


Figura 1. À esquerda, as cem primeiras iterações da série temporal em \hat{x} do mapa de Hénon e, à direita, o atrator correspondente à simulação

2.1.2. Mapa logístico

Conforme foi dito anteriormente, o outro sistema caótico a tempo discreto simulado foi o mapa logístico. Descrito em 1976 por Robert May, o mapa logístico é uma das formas de modelar a população de uma determinada espécie em certos instantes de tempo [8]. A equação de diferenças que descreve esse sistema pode ser vista abaixo:

$$x[n+1] = r \cdot x[n] \cdot (1 - x[n]) \quad (2)$$

Nesse caso, o sistema não chega a operar em caos para qualquer valor de r . Assim, como o estudo visa analisar o desempenho para sistemas caóticos, foi utilizado $r = 3.86$, que, conforme será visto no diagrama de bifurcação abaixo, faz com que a série temporal dada pela equação (2) opere em caos. Novamente, foram simuladas 5000 iterações:

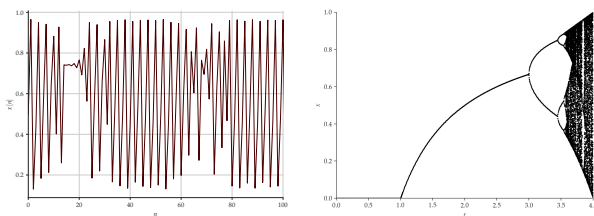


Figura 2. À esquerda, as cem primeiras iterações da série temporal do mapa logístico e, à direita, o diagrama de bifurcação deste sistema

2.1.3. Sistema de Lorenz

O sistema de Lorenz foi um dos sistemas dinâmicos caóticos a tempo contínuo simulados nessa pesquisa. Este sistema

foi um dos primeiros grandes trabalhos de sistemas caóticos, sendo considerado por muitos a pesquisa que inaugurou a área [11]. Lorenz modela, através de três equações diferenciais, o fluxo de um fluido em um volume uniformemente aquecido na camada inferior, e uniformemente resfriado na camada superior [9]:

$$\frac{dx}{dt} = -\sigma \cdot (x - y) \quad (3a)$$

$$\frac{dy}{dt} = x \cdot (\rho - z) - y \quad (3b)$$

$$\frac{dz}{dt} = x \cdot y - \beta \cdot z \quad (3c)$$

Para a simulação numérica foi considerado que $\sigma = 10$, $\beta = \frac{8}{3}$, $\rho = 28$ e foi utilizado $dt = 0.01$, gerando 5000 dados, assim como nos casos discretos. As figuras abaixo mostram a série temporal em \hat{x} e o atrator de Lorenz para a condição inicial $[x(0) \ y(0) \ z(0)]^T = [0.1 \ 0 \ 0]^T$:

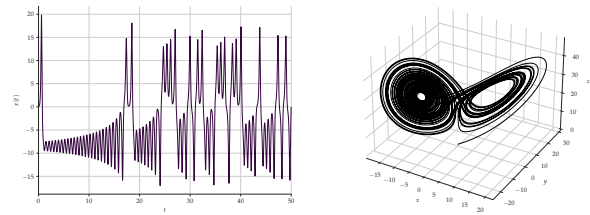


Figura 3. À esquerda, a série temporal em \hat{x} do sistema de Lorenz simulado e, à direita, o diagrama de fases correspondente à simulação

2.1.4. Equações de Mackey-Glass

Por fim, o último sistema caótico simulado, este também a tempo contínuo, foram as equações de Mackey-Glass. Essas equações diferenciais com atraso temporal modelam o controle hormonal da produção de células brancas do sangue e podem ser vistas abaixo [10]:

$$\frac{dP(t)}{dt} = \frac{\beta_0 \cdot \theta^n}{\theta^n + P(t-\tau)^n} - \gamma \cdot P(t) \quad (4a)$$

$$\frac{dP(t)}{dt} = \frac{\beta_0 \cdot \theta^n \cdot P(t-\tau)}{\theta^n + P(t-\tau)^n} - \gamma \cdot P(t) \quad (4b)$$

Neste caso, a equação (4b) exibe um comportamento caótico para valores mais altos de τ . Para a simulação numérica, foi utilizado $n = 10$, $\gamma = 0.1$, $\beta = 0.2$, $\theta = 1$, $\tau = 22$ e $dt = 1.0$, gerando novamente 5000 dados, resultando na seguinte série temporal e no seguinte atrator:

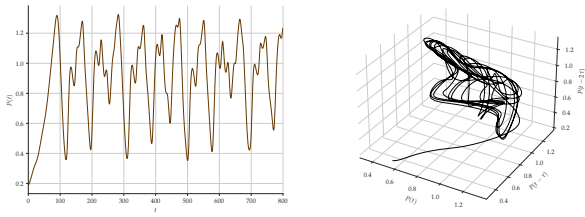


Figura 4. À esquerda, a série temporal da equação (4b) exibida de $t = 0$ a $t = 800$ e, à direita, o atrator correspondente à simulação

2.2. Obtenção dos melhores parâmetros

Como foi mencionado na seção introdutória, foi realizado um procedimento de *gridsearch* para a determinação dos melhores parâmetros para as redes neurais. Nesse caso, cada arquitetura testou parâmetros diferentes nesse processo, logo, as próximas seções indicam o que foi testado para cada modelo. Vale reforçar que o *gridsearch* foi realizado em todos os cenários de sistemas caóticos analisados, de forma a obter os melhores parâmetros para cada um deles.

2.2.1. MLP

Para as redes *Multilayer Perceptron*, foi testado o tamanho do *batch* para o treinamento (2, 4, 8, 16, 32), se será ou não utilizado uma camada de *batch normalization*, a função de ativação para os neurônios *Perceptron* na camada intermediária (SELU, ReLU, ELU, tanh, Sigmoid), a inicialização dos pesos dos neurônios (Glorot, He, Lecun, tanto com distribuição normal, quanto uniforme), o número de neurônios na camada intermediária (5, 10, 15, 20, 30, 50, 75, 100) e, por fim, a taxa de aprendizagem (0.001, 0.003, 0.005, 0.008, 0.01).

O *gridsearch* utilizou validação cruzada com 4 *folds* nos dados de treinamento, que correspondiam a 85% dos primeiros dados das séries temporais, isto é, os primeiros 4250 valores.

Vale reforçar que foi utilizada apenas uma camada intermediária em cada cenário. As melhores configurações obtidas podem ser vistas na tabela abaixo:

Cenário	Batch normalization	Batch size	Função de ativação	Inicialização	Nº de neurônios	Taxa de aprendizagem
Mapa de Hénon	Não	8	Sigmoid	Glorot Normal	50	0.003
Mapa logístico	Não	2	tanh	Glorot Uniforme	10	0.003
Sistema de Lorenz	Não	2	SELU	Lecun Normal	50	0.001
Equações de Mackey-Glass	Não	4	tanh	Glorot Normal	5	0.001

Tabela 1. Melhores parâmetros para a rede MLP nos cenários em análise

2.2.2. LSTM e GRU

O processo para a LSTM e GRU foram bem similares, considerando que essas arquiteturas de redes neurais são semelhantes. A principal diferença com relação ao *gridsearch* da

MLP foi o fato de que, ao invés de utilizar-se validação cruzada com *k-folds*, foi utilizado *holdout*. Esse processo dividiu o conjunto de treinamento (novamente composto por 85% dos dados gerados) em 4 seções. Cada seção era composta por uma fração dos dados de treinamento, sendo que cada seção incluía a seção anterior no seu conjunto de dados.

Por exemplo, a segunda seção obtida pelo *holdout* inclui a primeira seção e mais alguns dados, além de uma subdivisão de validação que será utilizada para avaliar o resultado. Com isso, obtém-se conjuntos sequenciais de dados para avaliação do desempenho. Esse procedimento é necessário para as redes recorrentes pois a relação temporal entre os dados de entrada deve ser levada em conta.

Foi avaliado o *batch size* (2, 4, 8, 16, 32), a inicialização dos pesos (Glorot Uniforme, Glorot Normal), o número de neurônios recorrentes na camada intermediária (5, 10, 15, 20, 30, 50, 75, 100), e a taxa de aprendizagem (0.001, 0.003, 0.005, 0.008, 0.01), novamente utilizando apenas uma camada intermediária. Vale reforçar que a função de ativação usual da célula recorrente não foi alterada (tanh), assim, para o sistema de Lorenz, foi feito um ajuste de escala para evitar a saturação na saída dos neurônios.

Os resultados obtidos para a LSTM e para a GRU podem ser vistos nas tabelas a seguir:

Cenário	Batch size	Inicialização	Nº de neurônios	Taxa de aprendizagem
Mapa de Hénon	4	Glorot Normal	15	0.005
Mapa logístico	2	Glorot Uniforme	100	0.008
Sistema de Lorenz	4	Glorot Uniforme	15	0.003
Equações de Mackey-Glass	2	Glorot Uniforme	50	0.003

Tabela 2. Melhores parâmetros para a rede LSTM nos cenários em análise

Cenário	Batch size	Inicialização	Nº de neurônios	Taxa de aprendizagem
Mapa de Hénon	4	Glorot Normal	30	0.003
Mapa logístico	2	Glorot Normal	100	0.003
Sistema de Lorenz	8	Glorot Uniforme	30	0.001
Equações de Mackey-Glass	2	Glorot Uniforme	10	0.005

Tabela 3. Melhores parâmetros para a rede GRU nos cenários em análise

2.2.3. ESN

A *Echo State Network* foi a rede neural mais distinta das outras analisadas nesta pesquisa. Essa rede recorrente baseia-se na teoria de computação por reservatório, chamada assim pois ela utiliza um reservatório de comportamentos dinâmicos que, ao serem combinados linearmente nos neurônios da camada de leitura, produzem as saídas da rede [12].

O grande diferencial prático desta rede é que os pesos dos neurônios do reservatório são ajustados com valores

fixos antes do treinamento da camada de saída. Esses valores são obtidos aleatoriamente, seguindo determinados parâmetros, como o raio espectral que determina o maior valor singular da matriz de pesos do reservatório, a taxa de vazamento que representa a velocidade com a qual o reservatório atualiza suas dinâmicas. Além disso, normalmente, a matriz dos pesos do reservatório é esparsa [4].

Devido a isso, o ajuste dos neurônios na camada de leitura pode ser realizado com um procedimento de regressão linear, sem necessitar de um treinamento iterativo, como nas redes anteriores. Esse fato economiza tempo de processamento e garante que será atingido o mínimo global da função custo utilizada.

Logo, para esta rede foi realizado um *gridsearch* para determinar o número de neurônios na camada de leitura (30, 50, 70, 90, 100, 120, 140, 160, 180, 200, 240, 280, 320, 360, 400, 440, 480, 500) e o valor para o raio espectral (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99) e, por tratar-se de uma rede recorrente, também foi utilizado o processo de *holdout* descrito na seção anterior. Vale também mencionar que a taxa de vazamento foi fixada em 0.9 para todos os cenários.

Obteve-se assim, as seguintes configurações para a ESN em cada um dos cenários:

Cenário	Raio espectral	Nº de neurônios
Mapa de Hénon	0.1	500
Mapa logístico	0.1	500
Sistema de Lorenz	0.2	120
Equações de Mackey-Glass	0.4	500

Tabela 4. Melhores parâmetros para a rede ESN nos cenários em análise

2.3. Análise comparativa

3. Resultados

4. Conclusões

Referências

- [1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [2] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [3] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.
- [4] H. Jaeger, "Echo state network," *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.
- [5] N. Fiedler-Ferrara and C. P. C. do Prado, *Caos: uma introdução*. Editora Blucher, 1994.
- [6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [7] M. Hénon, "A two-dimensional mapping with a strange attractor," *Communications in Mathematical Physics*, vol. 50, pp. 69–77, feb 1976.
- [8] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, pp. 459–467, jun 1976.
- [9] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.
- [10] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
- [11] J. Gleick, *Chaos: The amazing science of the unpredictable*. Vintage Publishing, 1998.
- [12] L. Boccato, *Novas propostas e aplicações de redes neurais com estados de eco*. Tese (doutorado), Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2013.